

# Visual Counting of Traffic Flow from A Car via Vehicle Detection and Motion Analysis

Kevin Kolcheck<sup>1</sup>, Zheyuan Wang<sup>1</sup>, Haiyan Xu<sup>2</sup> and Jiang Yu Zheng<sup>1</sup>

<sup>1</sup> Indiana University Purdue University Indianapolis, Indianapolis IN 46202, USA

<sup>2</sup> Fukuoka Institute of Technology, Fukuoka, Japan  
jzheng@iupui.edu xu@fit.ac.jp

**Abstract.** Visual traffic counting so far has been carried out by static cameras at streets or aerial pictures from sky. This work initiates a new approach to count traffic flow by using populated vehicle driving recorders. Mainly vehicles are counted by a camera moves along a route on opposite lane. Vehicle detection are implemented in video frames by using deep learning YOLO3 first, and then vehicle trajectories are counted in the spatial-temporal space called *motion profile*. Motion continuity, direction, and detection missing are considered to avoid multiple counting of oncoming vehicles. This method has been tested on naturalistic driving videos lasting for hours. The counted vehicle numbers can be interpolated as a flow of opposite lanes from a patrol vehicle for traffic control. The mobile counting of traffic is more flexible than the traffic monitoring by cameras at street corners.

**Keywords:** Computer Vision, Vehicle detection, Motion analysis, YOLO3, Traffic flow.

## 1 Introduction

Understanding traffic flow by counting number of vehicles is important for traffic control and daily life in a smart city. Current traffic counting relies on wired press sensors on road or GPS of vehicles transmitted via cellular network to map services [1]. The former method is inconvenient to be set on every street. While the latter method based on measuring vehicle speed can find traffic jams but not traffic volume because not every vehicle is communicating with the servers at all time. For the visual counting of traffic, CCTV at street corners also counts number of vehicles [2] at static locations. Another way is to fly a helicopter to capture flow at busy locations [3, 4]. All these methods require some setting of infrastructures. Our new approach uses a camera on a patrol vehicle or driving recorders on many cars traveling on streets [5]. We are attempting to obtain traffic flow information from these mobile cameras with the minimum sensing and computing costs, and then collect compact data via communication network. It is very common that a car driving outside witnesses traffic situations on the road. Compared to traffic monitoring CCTV at limited locations such as intersections, a mobile camera will not miss any accident on a route in the opposite direction and give a complete traffic report on its path. Multiple vehicles can count traffic flows on the

---

This is the author's manuscript of the article published in final edited form as:

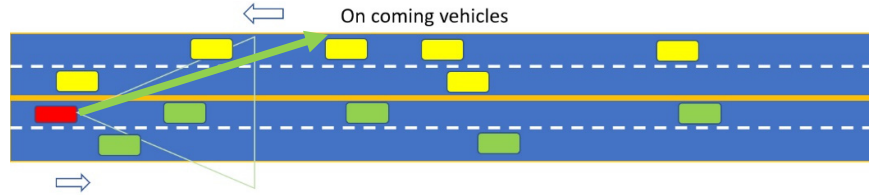
Kolcheck, K., Wang, Z., Xu, H., & Zheng, J. Y. (2020). Visual Counting of Traffic Flow from a Car via Vehicle Detection and Motion Analysis. In S. Palaiahnakote, G. Sanniti di Baja, L. Wang, & W. Q. Yan (Eds.), *Pattern Recognition* (pp. 529–543). Springer International Publishing. [https://doi.org/10.1007/978-3-030-41404-7\\_37](https://doi.org/10.1007/978-3-030-41404-7_37)

opposite roads at different times, and data are sent back to the map services. In general, your car is viewed by other cars on road more frequently than by sensors set on streets.

In driving video taken by a camera at the dashboard of a vehicle, we can recognize vehicles on all the lanes of current street by using YOLO3 (You Only Look Once) [6,7,8] with high detection accuracy. We focus the vehicle appearing regions of the opposite traffic in order to minimize the detection speed further on YOLO3. YOLOv3 real-time object detection system utilizes a single neural network to process a full image or frame. By comparison to classifier-based systems like R-CNN and Fast R-CNN, YOLO3 is much faster.

The YOLO3 detected vehicles have no detailed properties (e.g., vehicle face) associated with their motion direction. We further apply motion properties of vehicles such as direction and velocity for vehicle counting without tracking YOLO3 results over-time. To count traffic, however, we need to focus on vehicles on opposite lanes against current moving camera, because traffic in the same direction with the camera can be seen only locally in a limited range. Instead of recognizing vehicle faces such as front, back, and side [16], this work classifies the vehicle motion with respect to the camera. For the motion information processing of driving video, Kilicarslan et. al has proposed the motion profile as a spatio-temporal map to examine the vehicle trajectories [9] for vehicle identification [10], collision alarming [11,12], and pedestrian detection [13,14]. A motion trace longer than optical flow provides the stable motion of targets and can be used for object recognition because of the mechanical motion of vehicles, pedestrians, and static background.

The proposed method here only looks at opposite road. Assuming ego-vehicle moves on left lane and observes driving scenes as illustrated in Fig. 1. Oncoming vehicles through a road segment are all scanned in a period. Moreover, we only count those vehicles visible by the camera. Vehicles at farthest lane may be occluded completely by closer ones during traffic jam and they will be omitted. As a result, we can count vehicle in a good accuracy along road sections excluding intersection, roundabout, and one-way road.



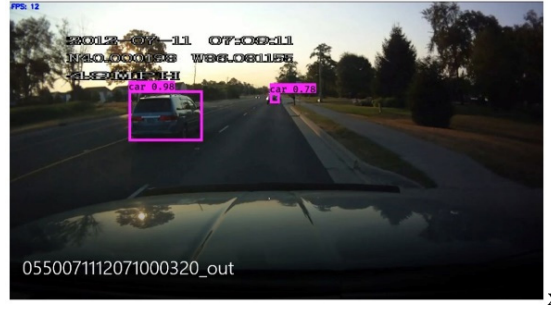
**Fig. 1.** A diagram to count vehicles on a route by a patrol vehicle (red) moving in the opposite direction. The observation vehicle can also be a normal car traveling on road. (yellow) oncoming vehicle from right to left, (green) vehicles moving in the same direction as the observing vehicle.

In the following section, we will first solve spatial recognition in each frame by using deep learning YOLO3 in Section 2. Section 3 shows the vehicle traces across video and the counting of vehicles in opposite lanes. Section 4 discusses the traffic estimation by mobile cameras vs other ways of traffic monitoring. Section 5 addresses experiments and section 6 gives a conclusion.

## 2 Vehicle Detection in Video Frame via YOLO3

As a vehicle moves forward, its forward camera can see surrounding vehicles moving in the same direction up to certain range, which is not effective for counting number of vehicles but can provide traffic flow information such as flow speed. On opposite lanes or road, the camera passes every oncoming vehicle and know the traffic flow along entire route. Oncoming vehicles appear from far away near the vanishing point of road and disappear on left margin as shown in Fig. 2. We can set the ROI only in left half of image to speed up the detection.

The default weights available on YOLO's website were trained on the COCO (Common Objects in Context) dataset and handle vehicle recognition out of the box. Among many objects YOLO3 can detect, vehicles are detected with a high accuracy viewed from street cameras and vehicle cameras. While YOLO3 does not provide any means of tracking objects from frame-to-frame, counting traffic is still possible beneficial from it due to YOLO's strong detection capabilities.



**Fig. 2.** A frame of driving video taken by a dashboard camera and detected vehicles by YOLO3 marked in bounding boxes.

The YOLO3 has been pretrained for detecting vehicles of various types such as car, truck, bus, etc. A bounding box is located approximately around an identified vehicle. The overall accuracy is over 90% on close vehicles without occlusion in our testing on naturalistic driving video. Even for partial occlusion between vehicles and by small obstacles such as fence and poles between the camera opposite roads, YOLO3 can still locate a box at the accuracy of 80% because of the underlying deep learning method looking at parts and their structure across different layers of the deep neural network. This maximumly helps the vehicle counting when an occlusion happens between vehicles on a wide road with multiple lanes. Figure 3 shows more results of vehicle detection under different illumination on several types of roads.

To count vehicles in opposition direction, we have to categorize all vehicles obtained by YOLO3 detection. Although the YOLO3 network can be retrained to find vehicle facing direction as for pedestrian facing direction [16], we employ the motion information to avoid a heavy training process of YOLO3, as the large dataset of vehicle samples have not been tagged with facing direction. Assume that ego-vehicle has a speed  $V_o$  and an observed vehicle has speed  $V$ . Their relative speed is then  $S=V-V_o$ .

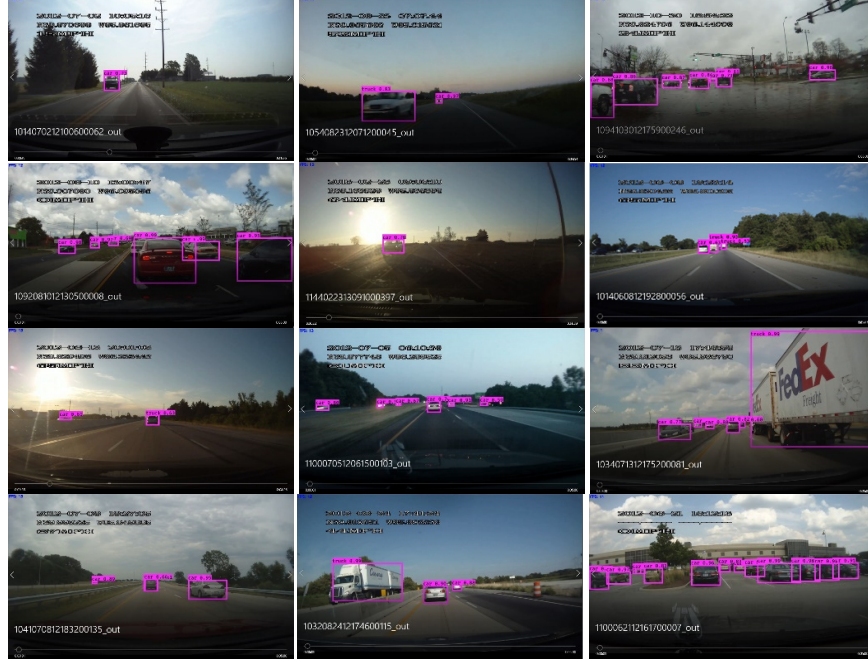


Fig. 3. Results of vehicle recognition by YOLO3 under different illumination conditions.

Different moving vehicles thus have the following motion properties.

*Front and parallel vehicles* ( $S \geq 0$ ): the front vehicle will not have long term negative relative speed in the visible range, otherwise it will cause a collision. Its position stays at frame center frequently. Multiple traces can be visible if more vehicles occupy several lanes and are moving in parallel to the camera. Their image velocity is low [10, 11].

*Passed vehicles* ( $S < 0$ ): If a vehicle with a lower speed than the ego-vehicle on a side lane is passed by the camera, the vehicle is moving in the video towards either side of the frame. The outward image velocity of such *passed vehicles* is larger than that of front vehicles.

*Passing vehicles* ( $S > 0$ ): If a *passing vehicle* overtakes the ego-vehicle or camera, its trajectory is moving from outside of video frame towards the frame center, because of the perspective projection of its speeding ahead motion. The inward image velocity has a different sign from that of a passed vehicle on the same side lane.

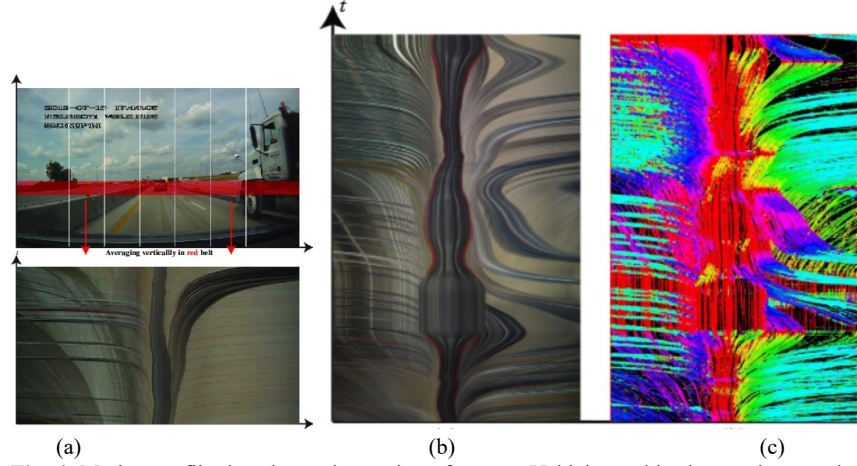
*Stopped vehicles* ( $S = -V_o$ ): for *stopped vehicles*, their image velocity is the same as those in static background, which diverges from the frame center [10].

*Oncoming vehicles* ( $S = -V - V_o < 0$ ): The fastest image velocity in negative  $x$  direction happens at *oncoming vehicles* on the opposite lane or road because the relative speed between oncoming vehicles and the moving camera is almost doubled. Oncoming vehicles only appear on left side of the video for right side driving countries.

### 3 Motion Trace Generation and Vehicle Counting

#### 3.1 Vehicles Motion Characteristics in Driving Video

To directly obtain motion of scenes, we generate a temporal image called *motion profile* [9] (MP) to record the trajectories of scenes in driving video. As seen in Fig. 4, a horizontal image belt (red) is located around the horizon projected into the video frame, which is almost invariant to the vehicle driving as long as the camera is set. All the vehicles, no matter how far or close, are covered by this belt completely or partially. We average the pixel color vertically in the belt  $B(x,y)$  and produce a line  $L(x)$ . The lines from consecutive frames form a spatio-temporal image  $M(x,t)$  called *motion profile* [9]. The motion profile provides more stable image velocity of scenes than between-frame optical flow. The motion can be measured locally from the orientation of traces computed by pixel differentiation, which avoid tracking to obtain motion in the 2D image sequence.

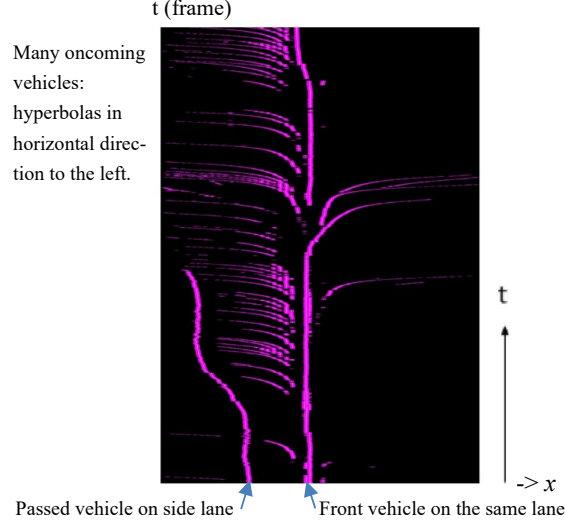


**Fig. 4.** Motion profile that shows the motion of scenes. Vehicles and background scenes have their motion traces appearing in the profile. The horizontal axis is the  $x$  axis of image and vertical axis is the time. The direction of motion traces can be calculated from the tangent of trace edges and colored accordingly. (a) frame belt and generation of motion profile, (b) a long motion profile with traces of vehicles, (c) traces in the motion profile are colored according to their orientation.

To ignore the motion from background scenes for vehicle counting. We plot YOLO detected vehicles only into another image aligned with the motion profile. For each bounding box  $b(x,y,w,h,t)$ , it is projected to an image  $m(x,t)$  as a line  $l(x,w,t)$ . The consecutive lines of bounding boxes form a trajectory in this spatio-temporal image, named *vehicle motion profile* (VMP) as shown in Fig. 5. The trajectory of a vehicle is more vertical in  $m(x,t)$ , if its horizontal image velocity is low in the video, while the trajectory is more horizontal (slanted) when the vehicle image velocity is large.

It is visible from the vehicle traces that YOLO3 can achieve more than 90% detection rate for close vehicles. The failure parts include the vehicles close to the margin of

view frame, because of the cut off vehicles by margin, a large occlusion of vehicle by some other vehicles at distance, and a far vehicle parked at roadside. Regardless of car overlap, the centroid of box is never overwritten if cars are detected.



**Fig. 5.** Vehicle motion profile that shows the vehicle trajectories. The horizontal lengths of bounding boxes are colored in pink in that frame and the box center is a white point.

The YOLO3 obtained boxes are not connected across frames. The maximum image velocity can be very large (negative) for oncoming vehicles. A general way to associate boxes along the time axis is to track boxes based on similar shapes of object. However, in our driving video, oncoming vehicle shape changes largely from front view near the image center (vanishing point) to sideview when it moves close to the frame margin. Moreover, the movement (image velocity) is large even if the video rate is 30 frame per second. If multiple vehicles are close to each other, they are easily confused because YOLO3 does not provide shape information in the output except the confidence value to be vehicles. Therefore, we focus on the local velocity of traces to avoid long term vehicle tracking to classify vehicles moving in different direction.

Let us examine the motion characters of vehicles around the moving camera. In the motion profile shown in Fig. 5, we can observe trajectories of detected vehicles along the time axis. These trajectories are from various vehicles. If the observing vehicle (ego-vehicle) is not moving on the leftist lane, it may also see stopped vehicles at roadside (parking, waiting for signal, in traffic jam, etc.). Also, the leftist lane may have vehicles passing the camera, or being passed by the camera, with their backside visible by the camera. They are all detected by YOLO3 without distinction in the left half of the video frame. To discriminate these vehicle traces from the oncoming vehicles, we examine their image velocities. Denote  $S$  be the relative speed of vehicle to the camera, and its position when captured is at  $(X, Z)$ , it is not difficult to derive the image velocity

$$v = \partial x / \partial t = \partial (fX/Z) / \partial t = -xfS/Z \quad (1)$$



where  $f$  is the camera focal length, the two vehicles are moving in opposite direction, and detailed derivation can be found in [10]. We can further find that the relative speeds of different cars are

$$\begin{aligned} S(\text{oncoming}) &< S(\text{stopping}) < S(\text{passed in same direction}) \\ &< S(\text{parallel in same direction}) = 0 < S(\text{passing in same direction}) \end{aligned} \quad (2)$$

Their image velocities have the relation as

$$\begin{aligned} v(\text{oncoming}) &< v(\text{stopping}) < v(\text{passed in same direction}) \\ &< v(\text{parallel}) = 0 < v(\text{passing in same direction}) \end{aligned} \quad (3)$$

If the ego-vehicle with camera stops, i.e.,  $V=0$ , the relation of image velocity becomes

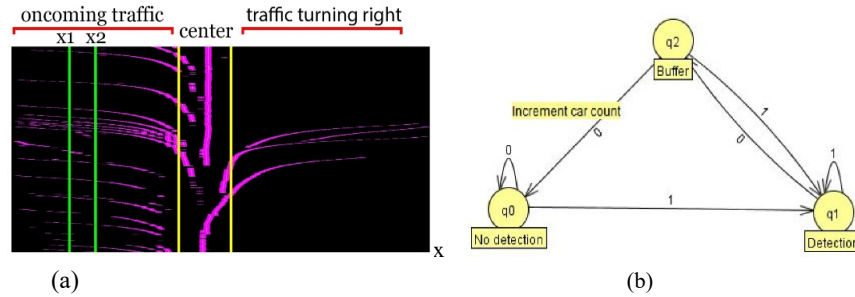
$$v(\text{on-coming}) < v(\text{stopping}) = 0 < v(\text{passing in same direction}) \quad (4)$$

These velocity constraints can be applied to relatively straight road only, but not at intersections or roundabouts where vehicles have no uniform motion direction and the traffic flow there has uncertain transition between road segments. Based on these constraints, we filter out oncoming vehicles from all other vehicles detected by YOLO3 in the vehicle motion profile using the image velocity obtained in the motion profile.

### 3.2 Counting Vehicles without Tracking Traces

How to count vehicles using trajectories? We have to either associate bounding boxes along the time axis as a unit, i.e., tracking vehicles, or just focus on a specific location in the frame for vehicle penetrating it. We employ the latter approach without tracking vehicles from frame to frame.

We set a vertical plane of sight from the camera in the 3D space as shown by the green arrow in Fig. 1. It scans passing vehicles on opposite road at front side. In this direction, the vehicle size is relatively stable with the scale variation according to the number of opposite lanes from the camera. By examining the motion profile, vehicle motion traces can be counted for the number of vehicles on the opposite lanes. This is possible because cars that appear in the opposite lane produce a distinct “hook” shape in the VMP where they slowly come into view in the distance and then quickly leave the frame as they get closer as in Fig. 6.



**Fig. 6.** Vehicle motion profile that shows the horizontal motion of vehicles on opposite lanes. The horizontal lengths of bounding boxes are colored in pink and the box center is a white point.

The implementation to count the distinct oncoming vehicles is to define a scanning zone with two boundaries  $x_1$  and  $x_2$  in the motion profile as shown in Fig. 6(a) for the purpose of identify each centroid that passes through. This fixed zone  $[x_1, x_2]$

- (1) is set toward left side as much as possible for the leftward moving direction of oncoming vehicles, but a little away from the frame margin where vehicles may not be detected due to their partial shape excluded in the frame;
- (2) is set away from the center of frame as much as possible where vehicles are small near the vanishing point. Also, this avoids the zone to cut a far road ahead possibly curved left, where the image velocity of vehicles becomes complex;
- (3) is set enough wide to cover the fastest vehicle velocity. We estimate the maximum image velocity of oncoming vehicles even if for the road curved right. The width  $(x_1, x_2)$  guarantees that the fastest oncoming vehicle has the centroid captured once in the zone.

Searching the zone from  $x_1$  to  $x_2$  starting from beginning along the profile, we count each centroid in each frame and comparing the number to previous frames. It is possible to count how many vehicles passing through  $[x_1, x_2]$ .

To filter motion traces with a certain velocity from  $v \ll 0$  to  $v < 0$ , we require the trace captured in the zone will not last too long, i.e., the duration of centroid within the zone is consecutively less than  $(x_1 - x_2)/v$  frames. Assume the number of centroids is  $n(t)$  at frame  $t$ . We count a vehicle in the total number  $N(t)$  when  $n(t)$  decreases (duration of a trace finishes). On the other hand, YOLO3 may still miss a frame or two along the trace due to various reasons including occlusion. A buffer of one frame is used so that, when the counted number of centroids in the zone changes relative to the previous frame, the total car count  $N(t)$  is not immediately incremented until there has been no detections for two frames further in a row. This accounts for missing frames where YOLO3 fails to make a detection. Additional detections after the first missing frame are resolved and added to the total car count  $N(t)$ , after the row count  $n(t)$  decreases from the previous frame count  $n(t-1)$ . We use an automaton to realize this counting as shown in Fig. 6(b).

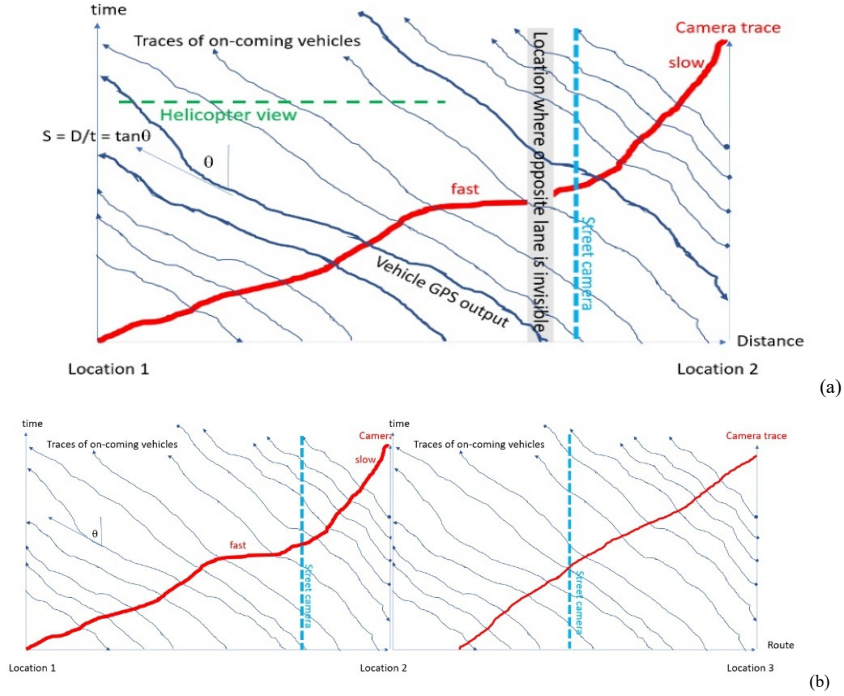
A benefit of doing detection based on centroids rather than when encountering expected color of bounding box in the motion profile is that, in busier videos, it is possible to make distinctions between multiple cars that pass by around the same time. Some occasional traces of passed vehicle may also go through the zone but in a much slower velocity, i.e.,  $v < 0$ ,  $|v|$  is small, and their trace is more vertically. But it will not add more count when its state keeps 1 in the automaton for too long. If ego-vehicle moves on the left lane, there is few chances that a passing vehicle will be observed on left side.

## 4 Dense Vehicle Counting for Traffic Flow Estimation

Here we introduce the nature of counting vehicles from mobile cars, and its potential application to traffic control. There is no method that can cover the real road traffic at the level of every car currently, except at some specific locations, at certain moment in a large area, or from a group of sampling cars. Figure 7(a) illustrates a traffic flow graph overtime [17] by depicting vehicle trajectories between two locations such as street intersections. According to different vehicle speeds, oncoming vehicles move from location 2 to location 1 (black traces). The observing vehicle (camera), however, moves



from location 1 to location 2 inversely and its trajectory is depicted as a red trace. It will go across those oncoming vehicles at curved traces. Our vehicle moves in distance  $d(t)$  on the road at time  $t$ . It counts the number of oncoming vehicles,  $N(d, t)$  on opposite lanes. To monitor a route with many road segments and intersections, we can concatenate the flow graphs of all segments along the route as in Fig. 7(b), with on-coming vehicle traces appearing and disappearing at intersection. The number of vehicles (traces) gives the density in the traffic flow graph and the tangent of traces indicates the traffic speed.



**Fig. 7.** Traffic flow chart changing over time along a road. The distance is along the horizontal axis and the time is vertically extended. (a) Traffic flow chart of one road segment. Oncoming vehicle trajectories are plotted in this distance-time space. (b) Traffic flow chart of consecutive road segments along a route. Two ego-vehicles are participating traffic flow counting. Oncoming vehicles may change the total number at street intersections (location  $i$ ) because of the split and merge of traffic flow.

A street camera monitoring a straight road obtain oncoming vehicles and their speed at different time, but the location is fixed as depicted by blue line in Fig. 7. A helicopter view is depicted in green line that captures a moment of road in a limited area. GPS based traffic monitoring, however, obtain multiple traces of vehicles equipped with GPS and wireless communication when the reporting function is turned on.

There are dense street cameras set in some cities for monitoring traffic flow, which has more blue vertical lines sampling the flow in Fig. 7. Through interpolation and

vehicle reidentification, a dense flow chart can be generated based on statistical models such as Bayesian Network [15]. GPS based traces can also provide a vehicle flow such as flow speed around a reporting vehicle due to the constraint of a smooth traffic flow. Aerial views are usually taken for a short while if a helicopter takes off. Nevertheless, a patrol vehicle on the opposite road can observe more locations on a longer route than static cameras and aerial photos, since a traffic flow has a slow and incremental change, e.g., a traffic jam may be diminished after 10 min to one hour. Our method to assign the counting task to more observing vehicles can also interpolate traffic flow between them. Ultimately, the red curves of cameras can be dense enough to interpolate entire traffic flow, when observing vehicles have short intervals in between.

Our method using driving recorders to count oncoming vehicles may miss counting when the road center has some obstacles, trees, and separator to block the views of opposite lane. A splitting of two-way road to lanes apart from each other such as in tunnels or on highway makes this situation. Our data to detect vehicles are missing at such a road segment. In this sense, the traffic flow chart in Fig. 7(a) may have some hidden spots on road that causes blind belts vertically extended in the spatio-temporal flow chart of traffic.

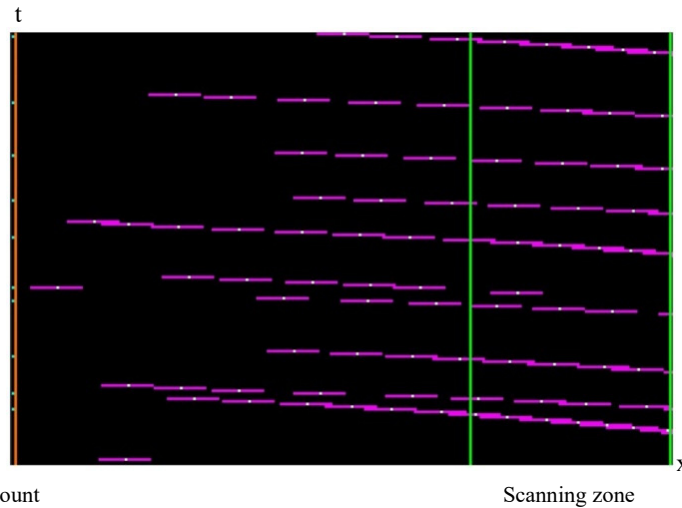
## 5 Experiments

### 5.1 Naturalistic Driving Data for Testing

For the experiment, we started with 500 5-minute long videos ( $1280 \times 720$ ) of dashcam footage from cars driving in a variety of weather, times of days, and road conditions. These videos were obtained from naturalistic driving on various roads in the City of Indianapolis [5]. The horizon of each video is manually specified once and then works for the entire clip. This allows us to locate ROI for car detection. We first sorted each video by road condition into three categories. The first is for videos of rural locations where the road is narrow, and traffic is low. The second category is local traffic, this would be busier locations, towns, etc. with roads that have more lanes than rural. The last category is highway and interstate footage. Some videos have overlap where they could technically fit in two categories, but we gave priority to which road type was most prominent. The videos were then subdivided into weather and time conditions [19], which are limited to sunny/cloudy, night, and rain/snow. We test our algorithm on naturalistic driving videos. Each clip results in 9000 frames in 30fps and they are passed to YOLO3 for vehicle detection. The detection achieved a high accuracy in sunny and cloudy days by using pretrained YOLO3 network.

YOLO3 is performed well on all sunny/cloudy videos but is not performed well on night or rainy/snowy footage. For the night footage, the issue is the default weights are not trained to recognize how vehicles look at nights, where cars are recognizable primarily by their headlights rather than vehicle shape like day-time video. The rainy and snowy footage was inconsistent due to water obscuring the windshield and concealing the shape of the vehicles, making frame to frame detections very spotty. After the motion profiles have been generated, we process them to count the motion traces present and export the results.

We processed the motion profiles in three separate batches, adjusting the size of the zone boundaries to widths 30pixels, 75pixels, and 125pixels. A few observations: for a video with the dimensions of  $1280 \times 720$ , we found that 30 pixels was not wide enough to reliably spot the majority of passing vehicles. Depending on the exact  $x$  coordinates, many cars would stop being detected before passing through or would be detected after passing the boundaries. For 125pixel zone width, on the other hand, was too large and more prone to counting cars multiple time due to a longer stretch of time required for a car to pass all the way through the boundaries, this made the counting algorithm more sensitive to inconsistent object detection. We found that 75pixel boundary was the proper spot and tended to yield more accurate counts, more often, as in Fig. 8.



**Fig. 8.** Partial Vehicle Motion Profile on the left shows the horizontal motion of vehicles on opposite lanes. The horizontal length of bounding boxes is colored in pink and the box center is a white point. Two green lines bound a zone to scan oncoming vehicles. White dots on left column with orange time axis are the vehicle counting yielded in green zone.

## 5.2 Counting Accuracy along A Route

The detected vehicles are displayed with the motion profiles and the time (frame number) is marked on side. We can observe the correctness of counting in 5 min driving for many clips. Overall, the accuracy is summarized in Table I. Recall, precision, and  $F_1$  score are presented in *italic* and **bold** figures in the table for three types of road respectively. Over 30 videos are selected from busy clips on opposite road among 500 videos for testing. Figure 9 shows long time counting from routes lasting 5 minutes each, which are also selected from two local, a highway, and a rural road with the highest traffic density including short stopping time before street intersection. These data further contain some complex situations passing road interactions, which can be ignored using GPS data and are not included in our evaluation.

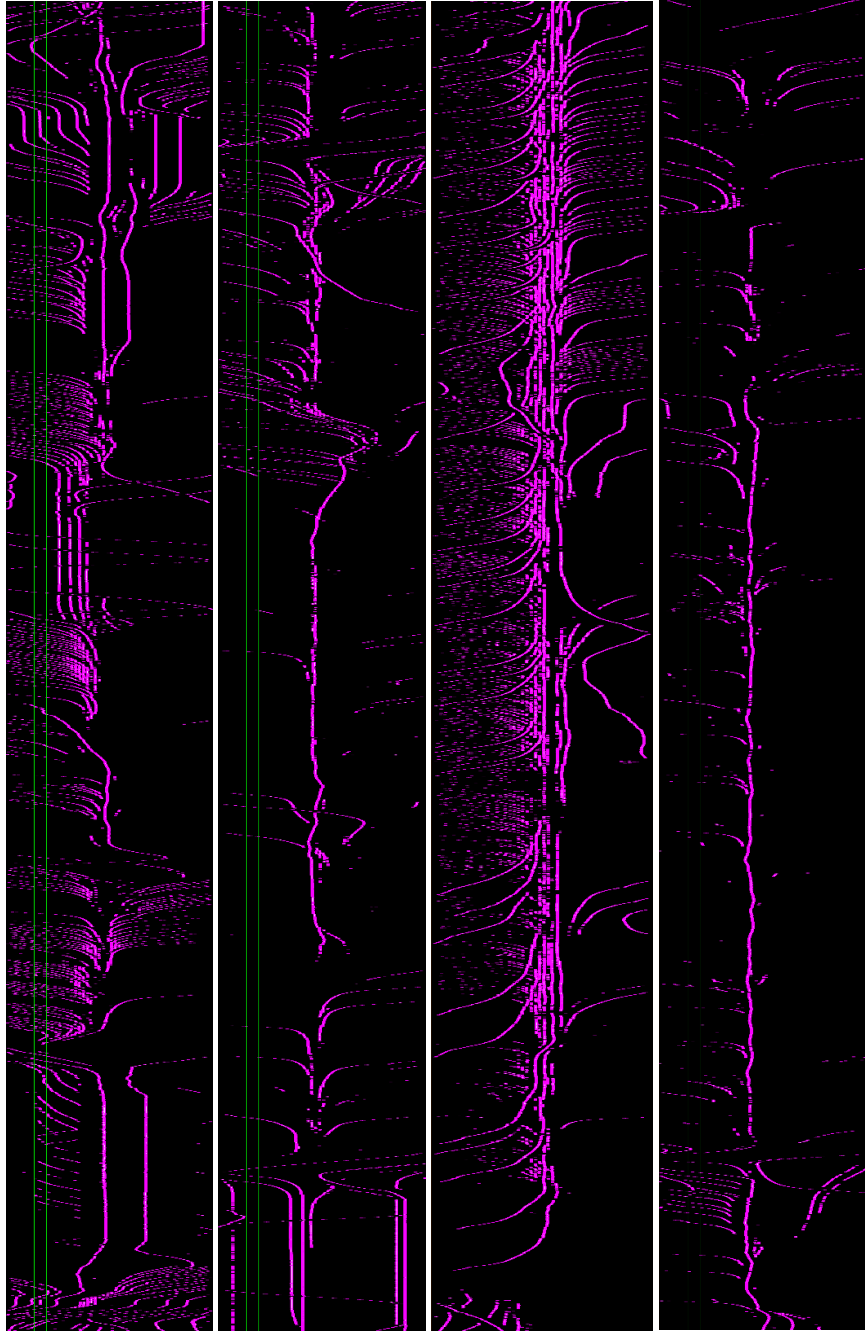


Fig. 9. Vehicle motion profiles showing oncoming vehicle traces and their extracted counts.

**Table 1.** Accuracy of Vehicle Counting. The digits indicate counted vehicle numbers

	Rural 10 video			Local 10 video			Highway 10 video		
	True	False	Recall	True	False	Recall	True	False	Recall
Detected	194	21	0.9	600	78	0.88	668	24	0.97
Undetected	6			37			134		
Precision	0.97		<b>0.93</b>	0.94		<b>0.91</b>	0.83		<b>0.89</b>

The false positive is a little large for local road because such roads also have parked cars in the opposite direction. The highway missed more vehicles than rural and local roads because of multiple lanes on separated opposite-road as well as possible occlusion of vehicles.

The real time consideration for the traffic flow counting demands us to reduce the computation cost and diminish the burden to the main tasks of camera such as road following, obstacle detection, and pedestrian avoidance. The following efforts can speed up the video.

- (1) YOLO3 selects bounding box candidates on a grid, which is the key mechanism of YOLO3.
- (2) Apply YOLO3 on left half of the frame, basically only around the belt and centered vertically around horizon.
- (3) If YOLO3 was trained on vehicle attributes [16] to identify subclasses of front and rare vehicles including front-side and rare-side, it could omit the time to distinguish traffic flow of oncoming vehicles and vehicles in same direction based on motion.

The future work could be training of YOLO3 to respond vehicles in night, on which only headlights are visible, while the motion characteristics are the same as daytime. We will further investigate the possibility to filter the vehicle traces directly in the motion profiles so that YOLO3 based vehicle recognition can be omitted; the vehicle counting is only based trace recognition.

## 6 Conclusion

This paper explores a new approach to detect traffic flow by counting vehicles from a traveling vehicle in contrast to the existing methods to monitor traffic at static locations by wire or viewed by cameras at streets, to fly helicopter over jammed area, or to gain GPS from a subset of vehicles in the traffic flow. The key components are the vehicle detection in dense video frames, and non-redundant vehicle counting based on motion properties of vehicles. The motion trace in the motion profile has a longer time behavior than optical flow so that we can focus on a location specific motion velocity to sort out opposite vehicle flow. We have achieved good accuracy of counting oncoming vehicles. We also reduced computing time of detection so that it can be deployed to many vehicles to perform this additional function aside the main task of ADAS or autonomous driving.

## References

1. W. Shi, Q-J. Kong, Y. Liu, A GPS/GIS integrated system for urban traffic flow analysis, IEEE International Conference on Intelligent Transportation Systems, 844-849, 2008.
2. S. Bagheri, J. Y. Zheng, S. Sinha: Temporal Mapping of Surveillance Video for Indexing and Summarization, Computer Vision and Image Understanding, vol. 144, 2016, 237-257.
3. S. Ram; J. Rodriguez: Vehicle detection in aerial images using multiscale structure enhancement and symmetry, International Conference on Image Processing (ICIP), 2016.
4. R. Ke, Z. Li, S. Kim, J. Ash, Z. Cui, Y. Wang: Real-time bidirectional traffic flow parameter estimation from aerial videos, IEEE Trans. Intelligent Transportation Systems, 18(4), 890-901, 2017.
5. R. Tian, L. Li, K. Yang, S. Chien, Y. Chen, R. Sherony, Estimation of the vehicle-pedestrian encounter/conflict risk on the road based on TASI 110-car naturalistic driving data collection. IEEE IV 2014, 623-629
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: You Only Look Once: Unified, Real-Time Object Detection, IEEE Computer Vision and Pattern Recognition (CVPR), 2016, 779-788.
7. J. Redmon, A. Farhadi, "YOLO: Better, Faster, Stronger" IEEE Conference on Computer Vision and Pattern Recognition, IEEE 2017.
8. J. Redmon, (n.d.). YOLO: Real-time object detection. Retrieved April 16, 2019, from <https://pjreddie.com/darknet/yolo>
9. M Kilicarslan, J. Y. Zheng: Visualizing driving video in temporal profile, Intelligent Vehicles Symposium, 2014 IEEE, 1263-1269.
10. A. Jazayeri, H. Cai, J. Y. Zheng, M. Tuceryan: Vehicle detection and tracking in car video based on motion model, IEEE Trans. Intelligent Transportation Systems, 12(2):583-595, 2011.
11. M. Kilicarslan, J. Y. Zheng: Predict vehicle collision by TTC from motion using a single video camera, IEEE Transactions on Intelligent Transportation Systems, 2018, 1-12.
12. Z. Gao, Y. Liu, J. Y. Zheng, R. Yu, X. Wang, P. Sun: Predicting hazardous driving events using Multi-Modal deep learning based on video motion profile and kinematics data, 21st International Conference on Intelligent Transportation Systems (ITSC), 2018.
13. M. Kilicarslan, J. Y. Zheng, K. Raptis, Pedestrian detection from motion, International Conference on Pattern Recognition, 1-6, 2016.
14. M. Kilicarslan, J. Y. Zheng, A. Algarni: Pedestrian detection from non-smooth motion, IEEE Intelligent Vehicle Symposium, 1-6, 2015.
15. T. A. Wheeler, M. J. Kochenderfer, and P. Robbel: Initial scene configurations for highway traffic propagation, 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, 279-284.
16. M. D. Sulistiyono, et. al.: Attribute-aware semantic segmentation of road scenes for understanding pedestrian orientations: IEEE International Conference on Intelligent Transportation Systems, 1-6, 2018.
17. Z. He, L. Zheng, L. Song, N. Zhu: A jam-absorption driving strategy for mitigating traffic oscillations, IEEE Trans. Intelligent Transportation Systems, 18(4), 802-813, 2017.
18. F. Porikli, X. Li: Traffic Congestion estimation using HMM models without vehicle tracking, IEEE Intelligent Vehicles Symposium, 2004.
19. G. Cheng, Z. Wang, J. Y. Zheng, Modeling weather and illuminations in driving views based on big-video mining, IEEE Trans. Intelligent Vehicles, 3(4), 522-533, Oct. 2018.